INSTITUT FÜR STATISTIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Seminar

–Regularisierungstechniken und strukturierte Regression–

Support Vector Machines

Autor: Giuseppe Casalicchio

Betreuer: Wolfgang Pößnecker

Abgabe: 8. März 2013

Inhaltsverzeichnis

1 Grundlagen

Bei der Klassifizierung durch Support Vector Machines geht man von N Trainingsdaten $(\mathbf{x}_1^\top, y_1), \ldots, (\mathbf{x}_N^\top, y_N)$ aus, wobei

$\mathbf{x}_i^\top \in \mathbb{R}^p$	ein Merkmalsvektor mit p Variablen und
$y_i \in \{-1, +1\}$	die Klassenzugehörigkeit der i-ten Beobachtung

ist. Die vorliegende Seminararbeit beschränkt sich auf die binäre Klassifizierung durch Support Vector Machines. Die grundlegende Idee besteht darin, die Daten durch eine Hyperebene in zwei Klassen aufzuteilen. Eine Hyperebene trennt dabei einen p-dimensionalen Variablenraum in zwei Halbräume und hat selbst die Dimension $(p-1)$. Die Gleichung einer Hyperebene hat die Form

$$\{\mathbf{x} \in \mathbb{R}^p \mid \mathbf{w}^\top \mathbf{x} + b = 0\}, \tag{1.1}$$

wobei $\mathbf{w} \in \mathbb{R}^p$ ein Vektor orthogonal zur Hyperebene und $b \in \mathbb{R}$ die Verschiebung (vom Ursprung) ist. Beispielsweise sind Hyperebenen

- in einem eindimensionalen Variablenraum alle Mengen, die aus einem Punkt bestehen,
- in einem zweidimensionalen Variablenraum alle Geraden und
- in einem dreidimensionalen Variablenraum alle Ebenen.

Das Ziel ist hierbei mit Hilfe einer Entscheidungsfunktion $f : \mathbb{R}^p \to \{-1, +1\}$ neu hinzukommende Beobachtungen $\mathbf{x}_{neu}$ möglichst fehlerfrei in die negative Klasse $y_{neu} = -1$ oder in die positive Klasse $y_{neu} = +1$ zuzuordnen. Die Entscheidungsfunktion wird auf Basis der Trainingsdaten so bestimmt, dass der Ausdruck $f(\mathbf{x}_i) = y_i$

- im Falle linear trennbarer Daten für alle Beobachtungen $i = 1, \ldots, N$ und
- im Falle nicht linear trennbarer Daten für möglichst viele Beobachtungen i

gilt (vgl. Schölkopf and Smola 2001, Kap. 7.1).

2 Linear trennbare Daten

Im Falle linear trennbarer Daten gibt es unendlich viele Hyperebenen, welche die zwei Klassen der Trainingsdaten vollständig voneinander trennen. Intuitiv betrachtet sind Hyperebenen zu bevorzugen, deren Abstand zu den Beobachtungen beider Klassen möglichst groß ist.

Die drei Hyperebenen in Abbildung 2.1 sollen diese Intuition veranschaulichen. Hier scheint die letzte Hyperebene "besser" als die ersten zwei zu sein, da der Abstand der Hyperebene zu beiden Klassen möglichst groß ist und man somit eher davon ausgehen kann, dass neu hinzukommende Beobachtungen richtig klassifiziert werden. Die Idee der Support Vector Machines ist, eine solche Hyperebene mit maximalen Abstand zwischen Beobachtungen beider Klassen zu finden. In den folgenden Kapiteln wird erklärt, wie diese Idee mathematisch umgesetzt werden kann.

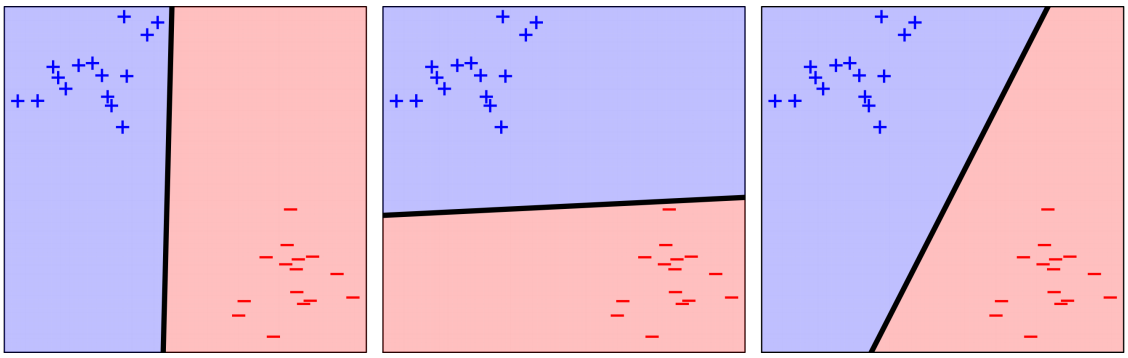

Abbildung 2.1: Mögliche Hyperebenen zur Trennung der Trainingsdaten

2.1 Entscheidungsfunktion

Zunächst wird begründet, wie sich die Entscheidungsfunktion $f(\mathbf{x})$ aus den Beobachtungen $\mathbf{x}$ zusammensetzt. Dazu betrachtet man in Abbildung 2.2 die zwei Halbräume

- $\mathbf{w}^\top \mathbf{x} + b > 0$ (blau) und
- $\mathbf{w}^\top \mathbf{x} + b < 0$ (rot),

die beim Trennen eines zweidimensionalen Variablenraums durch eine Hyperebene wie in Gleichung (1.1) entstehen. Mit Hilfe einer Entscheidungsfunktion $f(\mathbf{x})$ sollen

- Beobachtungen der positiven Klasse $y_i = +1$ in den Halbraum $\mathbf{w}^\top \mathbf{x} + b > 0$ und
- Beobachtungen der negativen Klasse $y_i = -1$ in den Halbraum $\mathbf{w}^\top \mathbf{x} + b < 0$

zugeordnet werden. Die Wahl von $f(\mathbf{x}) = \text{sgn}(\mathbf{w}^\top \mathbf{x} + b)$ ermöglicht eine solche Zuordnung (vgl. Ben-Hur and Weston 2010, Kap. 2; Schölkopf and Smola 2001, Kap. 7.1).

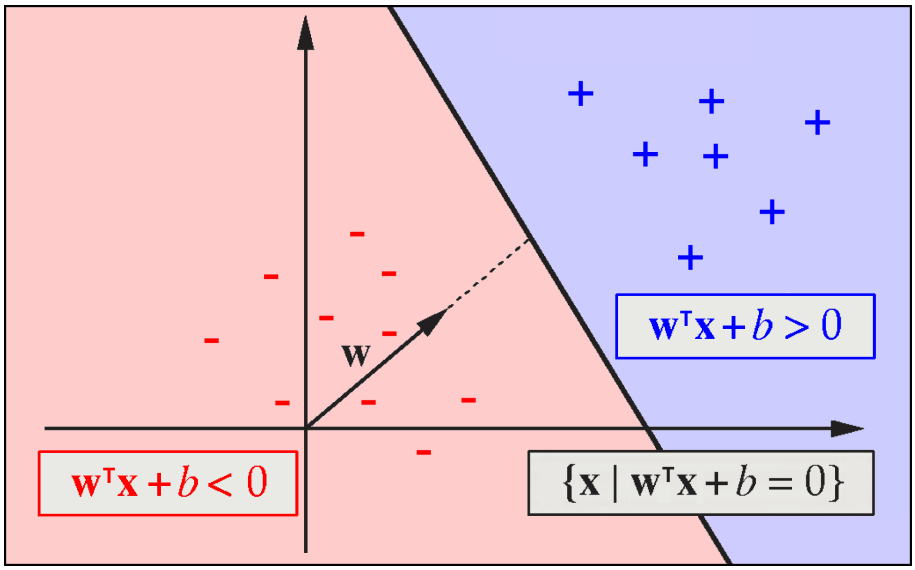

Abbildung 2.2: Bereiche “oberhalb” und “unterhalb” der Hyperebene

2.2 Kanonische Hyperebene

Die Hyperebene aus Gleichung (1.1) ist invariant gegenüber der Multiplikation mit einer beliebigen positiven Zahl λ. Die Lage und Position der Hyperebene bleibt wegen

$$\mathbf{w}^\top \mathbf{x} + b = 0 \;\Leftrightarrow\; \lambda \mathbf{w}^\top \mathbf{x} + \lambda b = 0,$$

für alle $\lambda \in \mathbb{R}^+$ unverändert. Daraus resultiert das Problem, dass es unendlich viele Gleichungen gibt, welche dieselbe Hyperebene beschreiben. Um dies zu umgehen, führt man eine sogenannte kanonische Hyperebene ein, die zusätzlich die Einschränkung

$$\min_{i=1,\ldots,N} |\mathbf{w}^\top \mathbf{x}_i + b| = 1$$

hat. Dadurch wird der Abstand der nächstgelegenen Beobachtungen zur Hyperebene so skaliert, dass er betragsmäßig gleich 1 ist. Dieser Abstand ist als funktionaler Abstand zu interpretieren, der lediglich die Abstände einer Beobachtung relativ zu den nächstgelegenen

Beobachtungen zur Hyperebene wiedergibt. Alle anderen Beobachtungen haben demnach einen betragsmäßigen funktionalen Abstand von der Hyperebene, der größer als 1 ist. Es gilt

$$\begin{cases} \mathbf{w}^\top \mathbf{x}_i + b \geq +1 & \text{für } y_i = +1 \\ \mathbf{w}^\top \mathbf{x}_i + b \leq -1 & \text{für } y_i = -1, \end{cases}$$

beziehungsweise in kompakter Form $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$. Diese Ungleichung gibt den betragsmäßigen funktionalen Abstand einer Beobachtung $\mathbf{x}_i$ orthogonal zur Hyperebene an (vgl. Schölkopf and Smola 2001, Kap. 7.1).

Abbildung 2.3 veranschaulicht diesen Sachverhalt am Beispiel eines zweidimensionalen Variablenraums: Die ausgefüllten Punkte auf den gestrichelten Geraden entsprechen den zur Hyperebene nächstgelegenen Beobachtungen. Ihr funktionaler Abstand orthogonal zur Hyperebene beträgt 1. Sie werden Stützvektoren (engl. *support vectors*) genannt, da sie den Richtungsvektor $\mathbf{w}$ der Hyperebene festlegen. Eine Verschiebung dieser Punkte bewirkt, dass sich der Richtungsvektor $\mathbf{w}$ und somit die Lage und Position der Hyperebene ändert (vgl. auch letzter Absatz in Kapitel 2.3). Die gestrichelten Geraden, auf denen die Stützvektoren liegen, definieren den sogenannten Rand (engl. *margin*). Durch Maximierung des Randes ermöglicht man, dass der Abstand der Hyperebene zu beiden Klassen möglichst groß wird. Im nächsten Kapitel ist das Ziel einen mathematischen Ansatz herzuleiten, der diesen Rand maximiert.

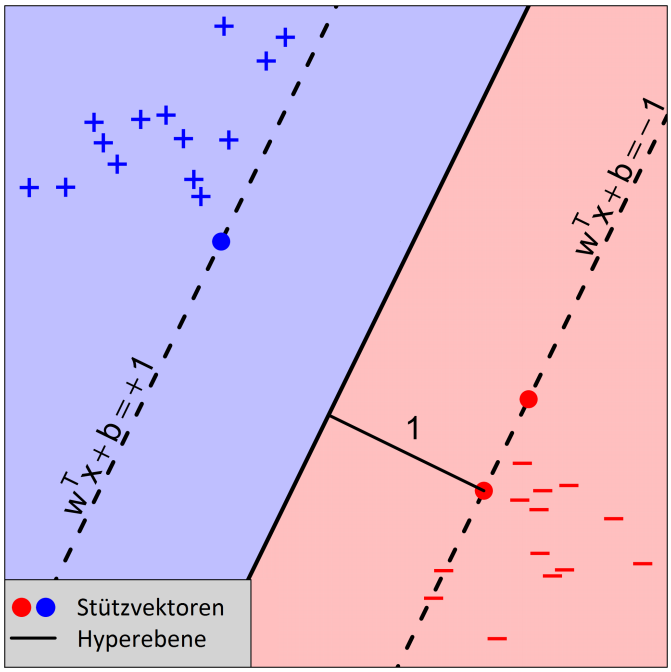

Abbildung 2.3: funktionale Abstand der Stützvektoren zur Hyperebene beträgt 1

2.3 Optimierungsproblem

Die Hyperebene wird durch die bisher unbekannten Parameter $(\mathbf{w}, b)$ festgelegt. Um den euklidischen Abstand einer Beobachtung $\mathbf{x}_i$ orthogonal zur Hyperebene zu bestimmen, muss zusätzlich mit der euklidischen Norm $||\mathbf{w}|| := \sqrt{\mathbf{w}^\top \mathbf{w}}$ normiert werden. Der euklidische Abstand berechnet sich durch

$$d((\mathbf{w}, b), \mathbf{x}_i) = \frac{y_i(\mathbf{w}^\top \mathbf{x}_i + b)}{||\mathbf{w}||} \geq \frac{1}{||\mathbf{w}||}.$$

Um diesen euklidischen Abstand und somit den Rand zu maximieren, genügt es $||\mathbf{w}||$ bzw. $\frac{1}{2}||\mathbf{w}||^2$ zu minimieren (vgl. Boswell 2002, Kap. 3).

Das daraus resultierende primäre Optimierungsproblem lautet

$$\begin{aligned} &\min_{\mathbf{w},b} \quad \frac{1}{2}||\mathbf{w}||^2 \\ &\text{NB:} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \quad \forall i = 1, \ldots, N. \end{aligned} \tag{2.1}$$

Die dazugehörige Lagrange-Funktion mit Lagrange-Multiplikatoren $\alpha_i \geq 0$ hat die Form

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{N} \alpha_i (y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1). \tag{2.2}$$

In der Literatur wird aus praktischen Gründen das duale Optimierungsproblem bevorzugt. Um dieses herzuleiten, wird die Lagrange-Funktion $L(\mathbf{w}, b, \boldsymbol{\alpha})$ bezüglich der Primärvariablen $\mathbf{w}, b$ minimiert und bezüglich $\boldsymbol{\alpha}$ maximiert (vgl. Schölkopf and Smola 2001, Kap. 7.3; Boyd and Vandenberghe 2004, Kap. 5).

Für das Minimieren bezüglich der Primärvariablen ergeben sich folgende Lösungen (vgl. Gunn et al. 1998, S. 8):

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial b} = 0 \Rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{2.3}$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \tag{2.4}$$

Setzt man Gleichung (2.3) und (2.4) in die Lagrange-Funktion (2.2) ein, erhält man die sogenannte Lagrange-duale Funktion

$$\begin{aligned} W(\boldsymbol{\alpha}) &= \min_{\mathbf{w},b} L(\mathbf{w}, b, \boldsymbol{\alpha}) = \min_{\mathbf{w},b} \left(\frac{1}{2}\mathbf{w}^\top \mathbf{w} - \sum_{i=1}^{N} \alpha_i (y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1) \right) \\ &\overset{(2.4)}{=} \min_{b} \left(\frac{1}{2} \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i^\top \sum_{j=1}^{N} \alpha_j y_j \mathbf{x}_j - \sum_{i=1}^{N} \alpha_i y_i \left(\sum_{j=1}^{N} \alpha_j y_j \mathbf{x}_j^\top \right) \mathbf{x}_i - \sum_{i=1}^{N} b \alpha_i y_i + \sum_{i=1}^{N} \alpha_i \right) \end{aligned}$$

$$
\begin{aligned}
&\overset{(2.3)}{=} \frac{1}{2}\sum_{i=1}^{N}\alpha_i y_i \mathbf{x}_i^\top \sum_{j=1}^{N}\alpha_j y_j \mathbf{x}_j - \sum_{i=1}^{N}\alpha_i y_i \left(\sum_{j=1}^{N}\alpha_j y_j \mathbf{x}_j^\top\right)\mathbf{x}_i + \sum_{i=1}^{N}\alpha_i \\
&= \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j - \sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j + \sum_{i=1}^{N}\alpha_i \\
&= -\frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j + \sum_{i=1}^{N}\alpha_i.
\end{aligned}
$$

Das duale Optimierungsproblem ist nicht mehr von den Primärvariablen $\mathbf{w}, b$ abhängig und lässt sich wie folgt aufstellen (vgl. Gunn et al. 1998, S. 8):

$$
\begin{aligned}
&\max_{\boldsymbol{\alpha}} W(\boldsymbol{\alpha}) = \max_{\boldsymbol{\alpha}} \left(\sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j\right) \\
&\text{NB:} \quad \alpha_i \geq 0, \qquad \sum_{i=1}^{N}\alpha_i y_i = 0, \qquad \forall i = 1, \ldots, N
\end{aligned}
\tag{2.5}
$$

Nach den Karush-Kuhn-Tucker (KKT) Bedingungen müssen alle Beobachtungen zusätzlich die Bedingung

$$\alpha_i[y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1] = 0$$

erfüllen (vgl. Schölkopf and Smola 2001, S. 197ff; Boswell 2002, S. 4). Dies impliziert, dass für alle Beobachtungen entweder $\alpha_i = 0$ oder $y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1$ gelten muss.

- $\Rightarrow$ $\alpha_i = 0$, wenn $y_i(\mathbf{w}^\top \mathbf{x}_i + b) > 1$, d.h. wenn der funktionale Abstand einer Beobachtung $\mathbf{x}_i$ zur Hyperebene größer als 1 ist.
- $\Rightarrow$ Es interessieren nur die Stützvektoren, da für diese $y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1$ gilt, d.h. der funktionale Abstand eines Stützvektors zur Hyperebene beträgt 1.

Für die Berechnung der Hyperebenenparameter $(\mathbf{w}, b)$ sind deshalb nur die Stützvektoren nötig. Für alle anderen Beobachtungen $\mathbf{x}_i$ sind die dazugehörigen $\alpha_i = 0$ und bleiben deshalb bei der Berechnung von $\mathbf{w} = \sum_{i=1}^{N}\alpha_i y_i \mathbf{x}_i$ unberücksichtigt. Somit ist der Parameter $\mathbf{w}$ lediglich eine Linearkombination der Stützvektoren. Der Parameter b kann aus der Gleichung

$$y_j = \mathbf{w}^\top \mathbf{x}_j + b = \sum_{i=1}^{N}\alpha_i y_i \mathbf{x}_i^\top \mathbf{x}_j + b \;\Leftrightarrow\; b = y_j - \sum_{i=1}^{N}\alpha_i y_i \mathbf{x}_i^\top \mathbf{x}_j$$

ermittelt werden. Die Entscheidungsfunktion zur Klassifizierung einer Beobachtung $\mathbf{x}_j$ lautet

$$f(\mathbf{x}_j) = \operatorname{sgn}\left(\mathbf{w}^\top \mathbf{x}_j + b\right) = \operatorname{sgn}\left(\sum_{i=1}^{N}\alpha_i y_i \mathbf{x}_i^\top \mathbf{x}_j + b\right).$$

3 Nicht linear trennbare Daten

3.1 Kern Trick

Bisher wurde angenommen, dass die Daten linear trennbar sind. Dies ist in der Praxis nicht immer der Fall. Eine Möglichkeit mit nicht linear trennbaren Daten umzugehen ist der sogenannte Kern Trick. Hierbei ist die Idee, die Daten mit Hilfe einer Funktion Φ in einen höherdimensionalen Variablenraum zu überführen, in dem sie linear trennbar sind.

Auf der linken Seite von Abbildung 3.1 ist ein solcher nicht linear trennbarer Fall in einem zweidimensionalen Variablenraum $\mathbf{x} = (x_1, x_2)$ dargestellt. Hierbei kann keine Hyperebene (hier Gerade) gefunden werden, so dass die roten Punkte von den blauen Kreuzen vollständig getrennt werden. Durch geschickte Wahl von Φ lassen sich die Daten in einen dreidimensionalen Variablenraum überführen, indem sie sich durch eine Hyperebene (hier Ebene) linear trennen lassen. Dies geschieht beispielsweise mit der Wahl von

$$\Phi(\mathbf{x}) = \Phi((x_1, x_2)) = (x_1^2, x_2^2, \sqrt{2}x_1x_2), \tag{3.1}$$

womit der zweidimensionale Variablenraum in einen dreidimensionalen Variablenraum überführt wird, d.h. $\Phi : \mathbb{R}^2 \to \mathbb{R}^3$. Die linear trennende Hyperebene im $\mathbb{R}^3$ hat die Form

$$\mathbf{w}^\top \Phi(\mathbf{x}) + b = 0 \tag{3.2}$$

$$\Leftrightarrow \mathbf{w}^\top (x_1^2, x_2^2, \sqrt{2}x_1x_2) + b = 0$$

$$\Leftrightarrow w_1x_1^2 + w_2x_2^2 + w_3\sqrt{2}x_1x_2 + b = 0. \tag{3.3}$$

Durch Einsetzen von Gleichung (3.1) in die Hyperebenengleichung (3.2) resultiert die Gleichung (3.3). Diese entspricht genau der Gleichung eines Kegelschnitts, was z.B. eine Hyperbel oder Ellipse im $\mathbb{R}^2$ sein kann (vgl. Ben-Hur and Weston 2010, Kap. 3). Durch die obige Wahl von Φ, können Daten, die im zweidimensionalen Variablenraum durch eine Ellipse trennbar sind, in einen dreidimensionalen Variablenraum durch eine Ebene linear getrennt werden. Die Wahl von Φ kann in der Praxis wegen Skalarproduktberechnungen im höherdimensionalen Raum der Form $\Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}_j)$ sehr rechenintensiv sein. Deshalb verwendet man häufig eine sogenannte Kernfunktion $K(\mathbf{x}_i, \mathbf{x}_j)$, welche die Funktion Φ implizit festlegt und meist auf Skalarproduktberechnungen im niedrigdimensionalen

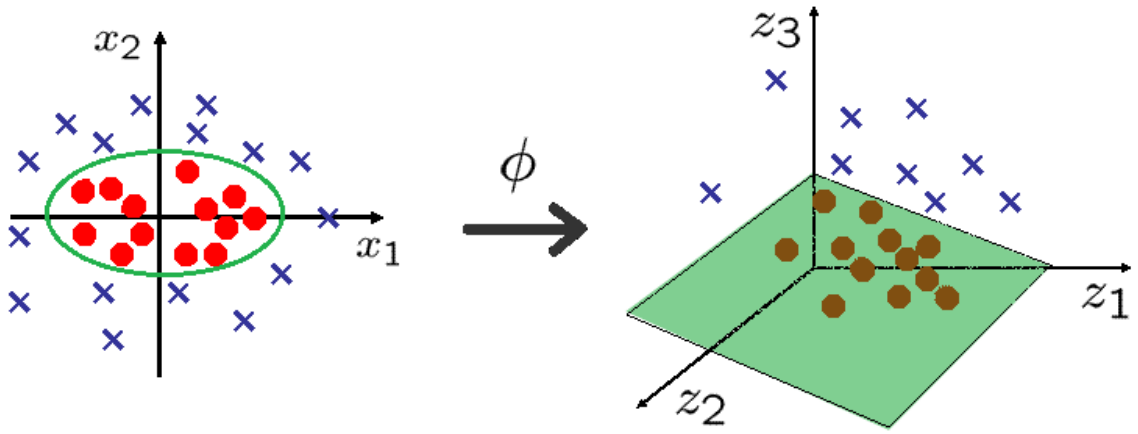

Abbildung 3.1: Überführung eines zweidimensionalen Variablenraums in einen höherdimensionalen (hier dreidimensionalen) Variablenraum, in dem die Daten linear trennbar sind. Dabei gilt $(z_1, z_2, z_3) := (x_1^2, x_2^2, \sqrt{2}x_1x_2)$.

Raum beruht. Beispielsweise beruht die polynomiale Kernfunktion auf Skalarproduktberechnungen der Form $\mathbf{x}_i^\top \mathbf{x}_j$. In der Literatur werden im Allgemeinen die folgenden Kernfunktionen genannt (vgl. Gunn et al. 1998, Kap. 3.1; Schölkopf and Smola 2001, Kap. 7.4):

- polynomiale Kernfunktion vom Grad d: $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j + c)^d$ mit $c \in \{0, 1\}$
- gaußsche radiale Basisfunktion: $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||}{c}\right)$, für $c > 0$.

Dabei sind c und d Kern-Parameter, die durch Kreuzvalidierung bestimmt werden können. Für eine Kernfunktion gilt stets

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}_j). \tag{3.4}$$

Das folgende Beispiel zeigt, dass eine polynomiale Kernfunktion vom Grad $d = 2$ mit $c = 0$ die Gleichung (3.4) erfüllt und zugleich die implizit festgelegte Funktion Φ identisch zu Gleichung (3.1) ist. Dabei werden Beobachtungen mit zwei Variablen, d.h. $\mathbf{x}_i = (x_{i_1}, x_{i_2})$, $i = 1, 2$ betrachtet:

$$\begin{aligned}
K(\mathbf{x}_1, \mathbf{x}_2) &= (\mathbf{x}_1{}^\top \mathbf{x}_2)^2 = ((x_{1_1}, x_{1_2})^\top (x_{2_1}, x_{2_2}))^2 \\
&= (x_{1_1}x_{2_1} + x_{1_2}x_{2_2})^2 \\
&= (x_{1_1}^2 x_{2_1}^2 + x_{1_2}^2 x_{2_2}^2 + 2x_{1_1}x_{1_2}x_{2_1}x_{2_2}) \\
&= (x_{1_1}^2, x_{1_2}^2, \sqrt{2}x_{1_1}x_{1_2})^\top (x_{2_1}^2, x_{2_2}^2, \sqrt{2}x_{2_1}x_{2_2}) \\
&= \Phi(\mathbf{x}_1)^\top \Phi(\mathbf{x}_2) \qquad \Rightarrow \Phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)
\end{aligned}$$

Ein Beweis, dass die Gleichung (3.4) auch für allgemeine polynomiale Kernfunktionen vom Grad d mit den Konstanten $c = 0$ bzw. $c = 1$ und für die gaußsche radiale Basisfunktion

gilt, kann in Schölkopf and Smola (2001, Kap. 2) bzw. Hastie et al. (2011, Kap. 5.8) nachgeschlagen werden.

Das weitere Vorgehen zur Bestimmung der Hyperebenenparameter $\mathbf{w}, b)$ ist analog zum Kapitel 2.3. Der einzige Unterschied ist, dass man den Merkmalsvektor $\mathbf{x}_i$ mit den höherdimensionalen Merkmalsvektor $\Phi(\mathbf{x}_i)$ ersetzt und bei Skalarproduktberechnungen der Form $\Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}_j)$ eine Kernfunktion $K(\mathbf{x}_i, \mathbf{x}_j)$ verwendet. Als Entscheidungsfunktion für eine Beobachtung $\mathbf{x}_j$ ergibt sich somit

$$f(\mathbf{x}_j) = \operatorname{sgn}\left(\sum_{i=1}^{N} y_i \alpha_i \Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}_j) + b\right) = \operatorname{sgn}\left(\sum_{i=1}^{N} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}_j) + b\right).$$

3.2 Soft Margin

Bisher wird durch die Nebenbedingung $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$ aus Gleichung (2.1), beziehungsweise beim Kern-Trick die Nebenbedingung $y_i(\mathbf{w}^\top \Phi(\mathbf{x}_i) + b) \geq 1$, gewährleistet, dass die Daten durch eine Hyperebene vollständig getrennt werden und man somit Fehlklassifizierungen vermeidet. Bei Ausreißern in den Daten kann dies zu Overfitting führen. Daher ist es wünschenswert Fehlklassifizierung zu erlauben, diese aber entsprechend zu bestrafen. In Abbildung 3.2 werden die Daten mit dem Kern Trick vollständig voneinander getrennt. Neue Beobachtungen auf der oberen linken Ecke würden hier in die negative (rote) Klasse klassifiziert werden, obwohl eine Klassifizierung in die positive (blaue) Klasse sinnvoller erscheint. Betrachtet man die grün eingekreisten Beobachtungen als Ausreißer, kann man deshalb hier von einem Overfittig ausgehen. Die Soft Margin Hyperebene in Abbildung 3.3 entschärft die oben erwähnte Nebenbedingung, so dass für einzelne Ausreißer eine Fehlklassifizierung erlaubt wird. Dies geschieht mit sogenannten Schlupfvariablen (engl. *slack variables*) $\xi_i \geq 0$. Die entschärfte Nebenbedingung lautet

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \; i = 1, \ldots, N$$

und ermöglicht, dass sich einige Beobachtungen innerhalb des Randes oder auf der falschen Seite der Hyperebene befinden dürfen. In Abbildung 3.3 sind das die mit $\xi_1, \ldots, \xi_5$ gekennzeichneten Beobachtungen. Dabei gibt ξ_i die Entfernung zwischen Beobachtung und der entsprechenden gestrichelten Linie an. Die Beobachtungen sind für

- $\xi_i = 0$ richtig klassifiziert,
- $0 < \xi_i \leq 1$ innerhalb des Randes richtig klassifiziert und
- $\xi_i > 1$ fehlklassifiziert (Ben-Hur and Weston 2010, Kap. 4.2).

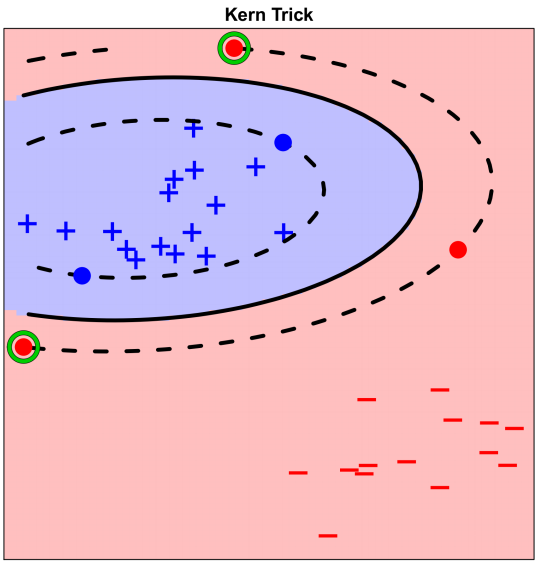

Abbildung 3.2: Overfitting beim Kern Trick

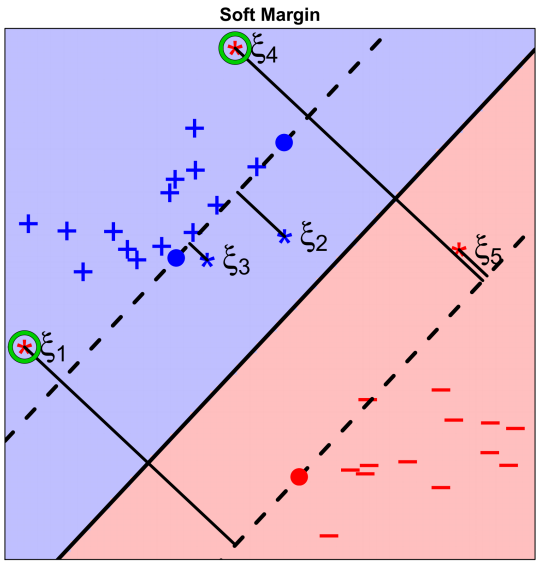

Abbildung 3.3: Zulassung von Ausreißern

Je breiter der Rand, desto mehr Beobachtungen liegen innerhalb des Randes oder auf der falschen Seite der Hyperebene. Die Summe der Schlupfvariablen $\sum_{i=1}^{N} \xi_i$ nimmt damit zu. Man möchte verhindern, dass diese Summe zu groß wird. Daher benötigt man einen Kompromiss zwischen dem Maximieren des Randes (bzw. $\min \frac{1}{2}||\mathbf{w}||^2$) und dem Minimieren von $\sum_{i=1}^{N} \xi_i$. Das Optimierungsproblem aus Kapitel 2.3 wird umformuliert zu

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N} \xi_i \\ \text{NB:} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \quad \forall i = 1, \ldots, N. \end{aligned}$$

Der Parameter C kann durch Kreuzvalidierung bestimmt werden und steuert wie stark die Summe $\sum_{i=1}^{N} \xi_i$ bestraft wird:

- bei großem C:
  Minimierung von $\sum_{i=1}^{N} \xi_i$ ist wichtiger → weniger Schlupfvariablen → kleiner Rand
- bei kleinem C:
  Maximierung des Randes ist wichtiger → mehr Schlupfvariablen → $\sum_{i=1}^{N} \xi_i$ größer

Die dazugehörige Lagrange-Funktion

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - (1 - \xi_i)) - \sum_{i=1}^{N} \mu_i \xi_i$$

mit Lagrange-Multiplikatoren $\alpha_i \geq 0$ und $\mu_i \geq 0$ wird bezüglich der Primärvariablen $\mathbf{w}, b$ und $\boldsymbol{\xi}$ minimiert und bezüglich $\boldsymbol{\alpha}$ maximiert. Es ergeben sich dieselben Lösungen wie in Gleichung (2.3) - (2.4) und die zur Minimierung bezüglich $\boldsymbol{\xi}$ hinzukommende Lösung

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu})}{\partial \boldsymbol{\xi}} = 0 \Rightarrow C = \alpha_i + \mu_i \quad \forall i = 1, \ldots, N. \tag{3.5}$$

Als Lagrange-duale Funktion folgt

$$\begin{aligned}
W(\boldsymbol{\alpha}) &= \min_{\mathbf{w},b,\boldsymbol{\xi}} L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) \\
&= \min_{\mathbf{w},b,\boldsymbol{\xi}} \left(\frac{1}{2}\mathbf{w}^\top \mathbf{w} \; + C \sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i (y_i(\mathbf{w}^\top \mathbf{x}_i + b) - (1 - \xi_i)) - \sum_{i=1}^{N} \mu_i \xi_i \right) \\
&= \min_{\mathbf{w},b,\boldsymbol{\xi}} \left(\frac{1}{2}\mathbf{w}^\top \mathbf{w} \; + \sum_{i=1}^{N} C\xi_i - \sum_{i=1}^{N} \alpha_i y_i \mathbf{w}^\top \mathbf{x}_i - \sum_{i=1}^{N} \alpha_i y_i b + \sum_{i=1}^{N} \alpha_i - \sum_{i=1}^{N} \alpha_i \xi_i - \sum_{i=1}^{N} \mu_i \xi_i \right) \\
&\overset{(3.5)}{=} \min_{\mathbf{w},b} \left(\frac{1}{2}\mathbf{w}^\top \mathbf{w} + \cancel{\sum_{i=1}^{N} (\alpha_i + \mu_i)\xi_i} - \sum_{i=1}^{N} \alpha_i y_i \mathbf{w}^\top \mathbf{x}_i - b \sum_{i=1}^{N} \alpha_i y_i + \sum_{i=1}^{N} \alpha_i - \cancel{\sum_{i=1}^{N} (\alpha_i + \mu_i)\xi_i} \right) \\
&\overset{(2.3)}{=} \min_{\mathbf{w}} \left(\frac{1}{2}\mathbf{w}^\top \mathbf{w} \; - \sum_{i=1}^{N} \alpha_i y_i \mathbf{w}^\top \mathbf{x}_i + \sum_{i=1}^{N} \alpha_i \right) \\
&\overset{(2.4)}{=} \frac{1}{2} \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i^\top \sum_{j=1}^{N} \alpha_j y_j \mathbf{x}_j - \sum_{i=1}^{N} \alpha_i y_i \left(\sum_{j=1}^{N} \alpha_j y_j \mathbf{x}_j^\top \right) \mathbf{x}_i + \sum_{i=1}^{N} \alpha_i \\
&= \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j - \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j + \sum_{i=1}^{N} \alpha_i \\
&= -\frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j + \sum_{i=1}^{N} \alpha_i .
\end{aligned}$$

Hier ist die Lagrange-duale Funktion identisch zum Fall linear trennbarer Daten (vgl. Kapitel 2.3). Das duale Optimierungsproblem

$$\max_{\boldsymbol{\alpha}} W(\boldsymbol{\alpha}) = \max_{\boldsymbol{\alpha}} \left(\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \right)$$

$$\text{NB:} \qquad 0 \leq \alpha_i \leq C, \qquad \sum_{i=1}^{N} \alpha_i y_i = 0 \qquad \forall i = 1, \ldots, N$$

hat im Vergleich zu Gleichung (2.5) die zusätzliche Nebenbedingung $\alpha_i \leq C$, die sich aus Gleichung (3.5) herleiten lässt: $C = \alpha_i + \mu_i \;\Leftrightarrow\; \alpha_i = C - \mu_i \overset{\mu_i \geq 0}{\Longrightarrow} \alpha_i \leq C$ (vgl. Ben-Hur and Weston 2010, Kap. 4.2; Gunn et al. 1998, Kap. 2.2).

Das weitere Vorgehen zur Bestimmung der Hyperebenenparameter $(\mathbf{w}, b)$ ist analog zum linear trennbaren Fall.

4 Anwendungsbeispiel

Das folgende Beispiel ist mit der statistischen Software `R` (R Core Team 2012) und dem Zusatzpaket `svmpath` (Hastie 2012) erstellt worden. Der dazugehörige R-Code befindet sich in der mitgelieferten Datei `Anwendungsbeispiel.R`. Eine weitere Umsetzung für Support Vector Machines in `R` ist unter anderem in der Funktion `svm()` aus dem Paket `e1071` (Meyer et al. 2012) zu finden.

Zunächst werden 200 Beobachtungen simuliert, die dann durch zufälliges Ziehen im Verhältnis von $4:1$ in Trainingsdaten und Testdaten aufgeteilt werden. Die $N = 160$ Trainingsdaten in Abbildung 4.1 sollen das Zwei-Spiralen-Problem in einem zweidimensionalen Variablenraum nachstellen. Dabei handelt es sich um ein bekanntes binäres Klassifikationsproblem, bei dem die zwei Klassen spiralförmig miteinander verzahnt sind.

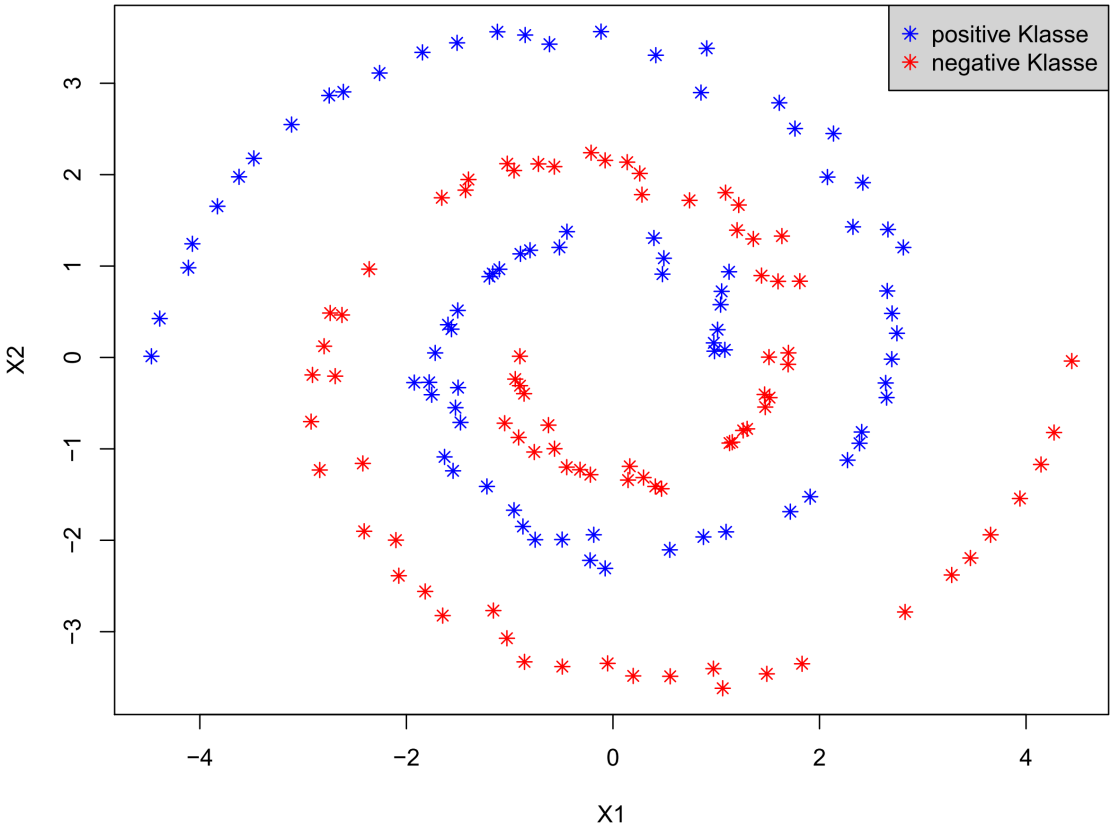

Abbildung 4.1: Spiralförmige Trainingsdaten

Gesucht ist eine Hyperebene, welche die Trainingsdaten für möglichst viele Beobachtungen voneinander trennt. Die Testdaten sollen anschließend, durch die auf Basis der Trainingsdaten geschätzten Hyperebene, möglichst fehlerfrei in die beiden Klassen zugeordnet werden.

In Abbildung 4.2 wurde die Hyperebene durch Verwendung einer polynomialen Kernfunktion zweiten Grades der Form $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j + 1)^2$ ermittelt. Im Vergleich dazu wurde in Abbildung 4.3 als Kernfunktion eine gaußsche radiale Basisfunktion (RBF) der Form $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-||\mathbf{x}_i - \mathbf{x}_j||)$ verwendet.

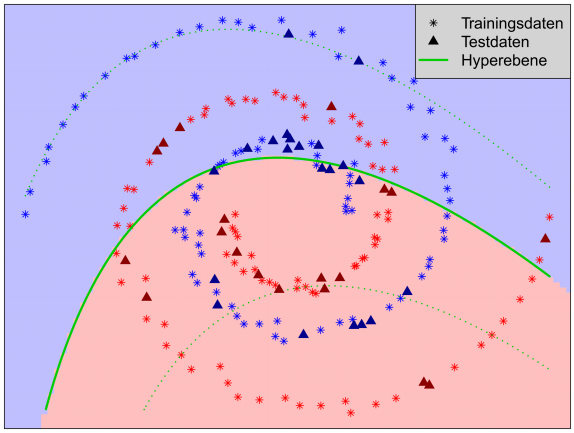

Abbildung 4.2: polynomiale Kernfunktion

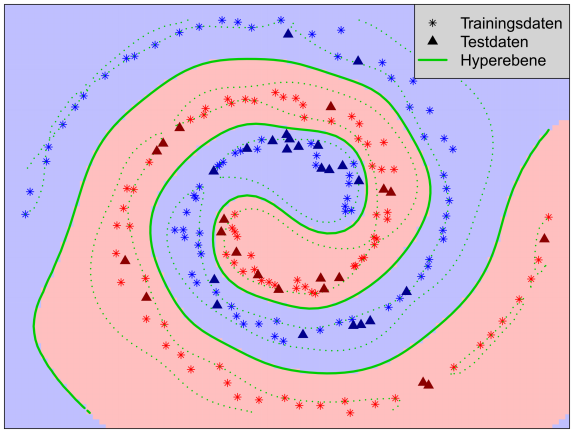

Abbildung 4.3: gaußsche RBF

Die polynomiale Kernfunktion kann die spiralförmig verzahnten Trainingsdaten hier nicht vollständig voneinander trennen, da die resultierenden Hyperebenen die Form eines Kegelschnitts haben (vgl. Kapitel 3.1). Die gaußsche radiale Basisfunktion hingegen überführt den Variablenraum in einen hochdimensionalen Variablenraum, bei dem eine Hyperebene gefunden werden kann, welches die Trainingsdaten vollständig voneinander trennt. Tabelle 4.1 zeigt dabei die Anzahl bzw. den Anteil der fehlklassifizierten Trainingsdaten und Testdaten.

	Trainingsdaten		Testdaten	
	Anzahl	Anteil	Anzahl	Anteil
Abbildung 4.2	67	41.875 %	16	40 %
Abbildung 4.3	0	0 %	0	0 %

Tabelle 4.1: Fehlklassifizierungen in Trainingsdaten und Testdaten

5 Zusammenfassung und Ausblick

Wenn das primäre Optimierungsproblem bekannt ist, lassen sich bei linear trennbaren Daten die Schritte zur Bestimmung der Hyperebenenenparameter $(\mathbf{w}, b)$ wie folgt zusammenfassen:

1. Lagrangefunktion aufstellen
2. Lagrange-duale Funktion und duales Optimierungsproblem herleiten
3. Lagrange-Multiplikator α_i durch duales Optimierungsproblem bestimmen
4. Richtungsvektor $\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i$ der kanonischen Hyperebene berechnen
5. Verschiebung $b = y_j - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i^\top \mathbf{x}_j$ der Hyperebene ermitteln
6. Entscheidungsfunktion $f(\mathbf{x}_j) = \operatorname{sgn}\left(\mathbf{w}^\top \mathbf{x}_j + b\right) = \operatorname{sgn}\left(\sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i^\top \mathbf{x}_j + b\right)$ aufstellen

Bei nicht linear trennbaren Daten ist eine Möglichkeit die Verwendung des Kern Tricks, der die Daten in einem höherdimensionalen Variablenraum überführt, in dem sie linear trennbar sind. Die Schritte zur Bestimmung der Hyperebenenenparameter ändern sich darin, dass der Merkmalsvektor $\mathbf{x}_i$ für alle $i = 1, \ldots, N$ durch einen höherdimensionalen Merkmalsvektor $\Phi(\mathbf{x}_i)$ ersetzt wird.

Die andere Möglichkeit ist die Verwendung der Soft Margin Hyperebene. Hierbei werden durch Einführung von Schlupfvariablen Fehlklassifizierungen ermöglicht, die zum Ausgleich im primären Optimierungsproblem entsprechend bestraft werden. Obwohl das primäre Optimierungsproblem anders als im linear trennbaren Fall ist, bleibt die Lagrange-duale Funktion dennoch dieselbe (vgl. Kapitel 3.2). Das duale Optimierungsproblem ändert sich darin, dass die zusätzliche Nebenbedingung $\alpha_i \leq C$ vorkommt. Die oben genannten Schritte können hier analog angewendet werden.

Zudem kann der Kern Trick auch bei der Soft Margin Hyperebene angewendet werden. Hierbei ersetzt man bei der Herleitung der Soft Margin Hyperebene den Merkmalsvektor $\mathbf{x}$ durch $\Phi(\mathbf{x})$.

In dieser Seminararbeit wurde nur die binäre Klassifikation betrachtet. In der Literatur findet man Erweiterungen, die die Verwendung von Support Vector Machines auf mehrkategoriale Klassifikationsprobleme und Regressionprobleme ermöglichen.

Ein Beispiel für den Umgang mit $K > 2$ Klassen ist die paarweise Klassifikation. Hierbei werden Klassifikatoren für jedes mögliche Paar der K Klassen gebildet und die Beobachtungen anschließend durch Mehrheitsentscheid in die Klasse $k \in \{1, \ldots, K\}$ zugeordnet (vgl. Hastie et al. 2011, Kap. 7.6).

Für die Erweiterung auf Regressionsprobleme muss das Optimierungsproblem zunächst in die **loss + penalty** Form gebracht werden. Dies geschieht indem man die Definition der Schlupfvariablen

$$\xi_i = \max\{0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)\} = [1 - y_i \underbrace{(\mathbf{w}^\top \mathbf{x}_i + b)}_{=:f(\mathbf{x}_i)}]_+$$

in ein leicht abgeändertes Optimierungsproblem aus Gleichung (3.2)

$$\min \ \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i \ \Leftrightarrow \ \min \ \frac{\lambda}{2}||\mathbf{w}||^2 + \sum_{i=1}^{N}\xi_i \text{ mit } \lambda = 1/C$$

einsetzt. Daraus ergibt sich folgende **loss + penalty** Form:

$$\min \ \sum_{i=1}^{N}[1 - y_i f(\mathbf{x}_i)]_+ + \frac{\lambda}{2}||\mathbf{w}||^2$$

Durch Verwendung der Schlupfvariablen impliziert man als Verlustfunktion den sogenannten **hinge loss**, der die Form $L(y_i, f(\mathbf{x}_i)) = [1 - y_i f(\mathbf{x}_i)]_+$ hat. Verwendet man eine andere Verlustfunktion $L(y_i, f(\mathbf{x}_i))$, können Support Vector Machines auch auf Regressionsprobleme angewendet werden (vgl. Gunn et al. 1998, Kap. 5.1; Hastie et al. 2011, Kap. 12.3.2).

Literaturverzeichnis

Ben-Hur, Asa and Weston, Jason (2010). A users guide to support vector machines. *Methods in Molecular Biology*, 609:223–239.

Boswell, Dustin (2002). Introduction to support vector machines.

Boyd, Stephen and Vandenberghe, Lieven (2004). *Convex optimization*. Cambridge university press.

Gunn, Steve R and others (1998). Support vector machines for classification and regression. *ISIS technical report*, 14.

Hastie, Trevor (2012). *svmpath: svmpath: the SVM Path algorithm*. R package version 0.952. Available from: http://CRAN.R-project.org/package=svmpath.

Hastie, Trevor and Tibshirani, Robert and Friedman, Jerome (2011). The elements of statistical learning, volume 18 of springer series in statistics.

Meyer, David and Dimitriadou, Evgenia and Hornik, Kurt and Weingessel, Andreas and Leisch, Friedrich (2012). *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien*. R package version 1.6-1. Available from: http://CRAN.R-project.org/package=e1071.

R Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0. Available from: http://www.R-project.org/.

Schölkopf, Bernhard and Smola, Alexander J (2001). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press.